

```

import React, { useState, useEffect, useCallback, useMemo } from "react";

import logo from './logo.svg';

import './App.css';

import Web3 from "web3";

// Contract address and ABI

const ADDRESS = "0xF72A4bfc0C5c1adA93fF97A81679f69cAbEA770E"; // Replace with your deployed
contract address

const ABI = [
  {
    "inputs": [
      { "internalType": "uint256", "name": "startingpoint", "type": "uint256" },
      { "internalType": "string", "name": "_startingMessage", "type": "string" }
    ],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "inputs": [],
    "name": "decreaseNumber",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "getNumber",
    "outputs": [{ "internalType": "uint256", "name": "", "type": "uint256" }],
    "stateMutability": "view",

```

```

    "type": "function"
  },
  {
    "inputs": [],
    "name": "increaseNumber",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  },
  {
    "inputs": [],
    "name": "message",
    "outputs": [{ "internalType": "string", "name": "", "type": "string" }],
    "stateMutability": "view",
    "type": "function"
  },
  {
    "inputs": [{ "internalType": "string", "name": "newMessage", "type": "string" }],
    "name": "setMessage",
    "outputs": [],
    "stateMutability": "nonpayable",
    "type": "function"
  }
];

```

```

function App() {
  const [number, setNumber] = useState("none");
  const [message, setMessage] = useState("");
  const [newMessage, setNewMessage] = useState("");

```

```
const [initialNumber, setInitialNumber] = useState("");
```

```
// Memoize the Web3 object
```

```
const web3 = useMemo(() => {  
  return new Web3(window.ethereum);  
}, []); // Only initialize once when the component mounts
```

```
// Memoize the myContract instance
```

```
const myContract = useMemo(() => {  
  return new web3.eth.Contract(ABI, ADDRESS);  
}, [web3]); // Depend only on the memoized web3
```

```
// Function to connect to MetaMask
```

```
const connectToMetaMask = async () => {  
  if (window.ethereum) {  
    await window.ethereum.request({ method: "eth_requestAccounts" });  
  } else {  
    alert("Please install MetaMask!");  
  }  
};
```

```
// Function to fetch the current number from the contract
```

```
const getNumber = useCallback(async () => {  
  try {  
    const result = await myContract.methods.getNumber().call();  
    setNumber(result.toString()); // Convert result to string  
  } catch (error) {  
    console.error("Error fetching number:", error);  
  }  
}
```

```
}, [myContract]);
```

```
// Function to fetch the current message from the contract
```

```
const getMessage = useCallback(async () => {
```

```
  try {
```

```
    const result = await myContract.methods.message().call();
```

```
    setMessage(result);
```

```
  } catch (error) {
```

```
    console.error("Error fetching message:", error);
```

```
  }
```

```
}, [myContract]);
```

```
// Function to set initial number
```

```
const handleSetInitialNumber = async () => {
```

```
  if (initialNumber && !isNaN(initialNumber)) {
```

```
    const accounts = await web3.eth.getAccounts();
```

```
    await myContract.methods.setMessage("Initial Number Set").send({ from: accounts[0] });
```

```
    await myContract.methods.increaseNumber().send({ from: accounts[0] });
```

```
    setInitialNumber(""); // Clear input field
```

```
    await getNumber(); // Update the number state
```

```
    await getMessage(); // Fetch the message after setting initial number
```

```
  } else {
```

```
    alert("Please enter a valid number.");
```

```
  }
```

```
};
```

```
// Function to increment the number
```

```
const increaseNumber = async () => {
```

```
  const accounts = await web3.eth.getAccounts();
```

```

    await myContract.methods.increaseNumber().send({ from: accounts[0] });

    await getNumber(); // Fetch updated number
};

// Function to decrement the number
const decreaseNumber = async () => {
    const accounts = await web3.eth.getAccounts();

    await myContract.methods.decreaseNumber().send({ from: accounts[0] });

    await getNumber(); // Fetch updated number
};

// Function to update the message
const updateMessage = async () => {
    const accounts = await web3.eth.getAccounts();

    await myContract.methods.setMessage(newMessage).send({ from: accounts[0] });

    setNewMessage(""); // Clear input field

    await getMessage(); // Update the message state
};

useEffect(() => {
    connectToMetaMask();

    getNumber(); // Fetch number when the component mounts
    getMessage(); // Fetch message when the component mounts
}, [getNumber, getMessage]); // Add functions as dependencies

return (
    <div className="App">
        <header className="App-header">
            <img src={logo} className="App-logo" alt="logo" />

```

```

    <button onClick={getNumber}>Get Number</button>

    <br />

    <p>Number: {number}</p>

    <button onClick={getMessage}>Get Message</button>

    <p>Message: {message}</p>

    <input
      type="number"
      value={initialNumber}
      onChange={(e) => setInitialNumber(e.target.value)}
      placeholder="Enter initial number"
    />

    <button onClick={handleSetInitialNumber}>Set Initial Number</button>

    <button onClick={increaseNumber}>Increase Number</button>

    <button onClick={decreaseNumber}>Decrease Number</button>

    <input
      type="text"
      value={newMessage}
      onChange={(e) => setNewMessage(e.target.value)}
      placeholder="Enter new message"
    />

    <button onClick={updateMessage}>Update Message</button>

  </header>

</div>

);
}

export default App;

```